

Zadanie 1

Zadanie składa się z części podstawowej (oznaczonej kolorem **czarnym**) oraz części rozszerzonej (oznaczonej kolorem **niebieskim**).

Treść

Zadanie polega na napisaniu REST API dla systemu logowania czasu.

Wymagania funkcjonalne

1. Z systemu mogą korzystać zalogowani **użytkownicy**. Zakłada się, że użytkownicy są uwierzytelnieni w systemie za pomocą JWT, który zawiera identyfikator użytkownika w formacie UUID. Nie jest wymagany żaden mechanizm rejestracji ani logowania, a jedynie uwierzytelnianie.
2. System umożliwia każdemu użytkownikowi założenie nowego **projektu**. Projekt składa się z:
 - a. Unikatowego identyfikatora projektu w formie tekstowej, podawanego przez użytkownika.
 - b. Znacznika czasu utworzenia projektu do celów ewidencjonowania.
3. System umożliwia autorowi projektu zmianę identyfikatora projektu.
4. System umożliwia każdemu użytkownikowi zalogowanie czasu spędzonego nad projektem (dalej jako **zadanie**). Zadanie składa się z:
 - a. Znacznika czasu rozpoczęcia pracy nad projektem.
 - b. Czasu trwania zadania.
 - c. Opcjonalnie: wolumenu wyrażonego liczbą naturalną.
 - d. Opcjonalnie: komentarz.
5. Dany użytkownik nie może zapisać dwóch zadań, które zachodzą na siebie w czasie według atrybutów tych zadań.
6. System umożliwia autorowi zadania usunięcie go, musi to być jednak "soft delete" opatrzony znacznikiem czasu usunięcia.
7. System umożliwia autorowi zadania zmianę wszystkich wyżej wymienionych atrybutów zadania przy zachowaniu walidacji z punktu 5. Powinno się to odbywać jako usunięcie zadania (jak w pkt 6) i stworzenie w jego miejsce nowego.
8. System umożliwia autorowi projektu usunięcie go, musi to być jednak "soft delete" opatrzony znacznikiem czasu usunięcia. Wszystkie istniejące zadania usuniętego projektu uznaje się za usunięte z tym samym znacznikiem czasu.
9. System umożliwia zwrócenie informacji o podanym projekcie wraz z przynależnymi zadaniami i sumarycznym czasem trwania projektu.
10. System umożliwia wylistowanie projektów wraz z przynależnymi zadaniami z zastosowaniem koniugacji następujących filtrów (każdy jest opcjonalny):
 - a. lista identyfikatorów,
 - b. od - znacznik czasowy utworzenia,

- c. do - znacznik czasowy utworzenia,
 - d. usunięte/nieusunięte,
- oraz sortowania z zastosowaniem co najwyżej jednego z kryteriów:
- e. znacznik czasowy utworzenia - malejąco/rosnąco,
 - f. znacznik czasowy aktualizacji - najpóźniej dodanego zadania (w przypadku pustego projektu - należy przyjąć datę utworzenia) - malejąco/rosnąco,
- oraz prostej paginacji opartej na rozmiarze i numerze strony.

11. System umożliwia wyświetlenie następujących statystyk (do statystyk liczą się tylko nieusunięte projekty i zadania):

- a. Łączna liczba zadań.
- b. Średni czas trwania zadania.
- c. Średni wolumen zadania (dla zadań z podanych wolumenem).
- d. Średni czas trwania zadania ważony wolumenem (dla zadań z podanych wolumenem).

Dostęp do statystyk powinien być parametryzowany listą identyfikatorów użytkowników, których zadania należy uwzględnić oraz datami od-do w postaci rok-miesiąc.

Wymagania niefunkcjonalne

1. Z systemu może korzystać wielu użytkowników jednocześnie.
2. Serwer aplikacyjny powinien działać poprawnie w środowisku rozproszonym - tzn. jednocześnie może być uruchomionych wiele instancji.
3. Tam gdzie to możliwe należy używać formatów ISO.
4. Statystyki powinny być zaprogramowane w sposób, który umożliwia ich pobieranie w zbliżonym czasie niezależnym od liczby zapisanych projektów i zadań.

Komentarz

- Myśl o systemie jakby miał być faktycznie używany w produkcji.
- Jeśli coś nie jest opisane z biznesowego punktu widzenia, to dopuszczona jest rozsądna dowolność.
- Zwróć szczególną uwagę na: jakość kodu; walidacje; odpowiednie zaprojektowanie endpointów REST-owych (albo REST-o-podobnych jeśli nie jesteś purystą) - ścieżki, metody (HTTP methods), odpowiednie kody HTTP, JSONy (także błędy zwracane w JSONie); rozsądne użycie typów; testy jednostkowe i integracyjne.